ARMY RESEARCH LABORATORY

Approaches to Health Monitoring of the CAT 7 Diesel Engine

by Andrew J. Bayba, David N. Siegel, Kwok Tom, and Canh Ly

ARL-TR-5677 **September 2011**

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory
Adelphi, MD 20783-1197

ARL-TR-5677 **September 2011**

Approaches to Health Monitoring of the CAT 7 Diesel Engine

Andrew J. Bayba, David N. Siegel, Kwok Tom, and Canh Ly
Sensors and Electron Devices Directorate, ARL

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED (From - To)
September 2011	Final	October 2010 to July 2011

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Approaches to Health Monitoring of the CAT 7 Diesel Engine	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)	5d. PROJECT NUMBER
Andrew J. Bayba, David N. Siegel, Kwok Tom, and Canh Ly	1NE6KK
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
U.S. Army Research Laboratory ATTN: RDRL-SER-E 2800 Powder Mill Road Adelphi MD 20783-1197	ARL-TR-5677

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This study consists of an evaluation of health assessment approaches for the CAT 7 diesel engine. The report presents the processing method and results from two potential approaches for monitoring ground vehicle engine health. This work is based on data acquired from seeded fault testing performed on the engine by the U.S. Army Tank and Automotive Research, Development and Engineering Center (TARDEC) from May–August 2010. Correlation analysis and principal component analysis were performed on the data and the results are presented. The results show detection of the presence of the majority of the faults. It is anticipated that through the application of various processing techniques such as these, a better assessment of engine health, in particular, the early detection of subsystems faults, is possible.

15. SUBJECT TERMS

Health monitoring, diesel engine health, CAT Y diesel engine, diesel engine prognostics and diagnostics

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Andrew J. Bayba
Unclassified	Unclassified	Unclassified	UU	34	19b. TELEPHONE NUMBER (*Include area code*) (301) 394-0440

Contents

List of Figures

List of Tables

INTENTIONALLY LEFT BLANK.

1. Introduction

There is a great deal of interest in the U.S. military in monitoring the health of assets in operation in the field. The primary motivation is so that timely, efficient, and effective decision making can be made both for operations and logistical support. With this in mind, the U.S. Army Research Laboratory (ARL) has teamed with the U.S. Army Tank and Automotive Research, Development and Engineering Center (TARDEC) to investigate approaches for assessing the health of diesel engines (Technology Program Annex TA-SE-2010-5). Seeded fault testing was executed with the assistance of Millennium Integrated Services (MIS) 2000/Global Defense under ARL contract W911NF-09-2-0036. The focus of this report is to present progress in this area, and specifically, to review the efficacy of algorithms that can detect anomalous conditions intentionally imposed on the system (seeded faults) and then identify the source of the variation that caused the anomaly. It is also anticipated that the effort in this particular subject will be applicable to other areas of interest in ARL's prognostics and diagnostics (P&D) program.

2. Experimental

A military version of the CAT 7 diesel engine (Model C7 DITA) was installed and instrumented in a dynamometer (dyno) test cell at TARDEC's facilities in Warren, MI (figure 1). The basics of the setup and data collected are described here; for a detailed description of the experiment, see reference 1. The setup was designed so that the engine could be operated and controlled without the presence a vehicle. The test stand supported provision of fuel, coolant, inlet air, and exhausting of the engine as well as a load (eddy current dyno, computer controlled). Data were collected from a variety of sources including existing sensors on the engine through the controller-area network (CAN) vehicle bus standard, several sensors in the test cell recorded by the cell data acquisition system (DAQ), and a few "add-on" sensors that were recorded at a higher rate (referred to as "analog data"). A small portion of the data that is referred to as "digital data" is primarily used for timing. There were also sensors inserted and data collected by the Pennsylvania State University (Penn State) Applied Research Laboratory. Both the CAN and dyno data were collected at a relatively low rate and provided to ARL at 1 Sample/s, and could be monitored continuously during a test run. The analog data were collected at 10 kiloSamples/s and, due to the high rate, "snapshots" of data of between 1 and 30 s were collected at select times during a test run. The Penn State data were collected independently without time synchronization of the TARDEC data at 102.4 kiloSamples/s. A diagram of the engine control, instrumentation, and data flow is shown in figure 2.

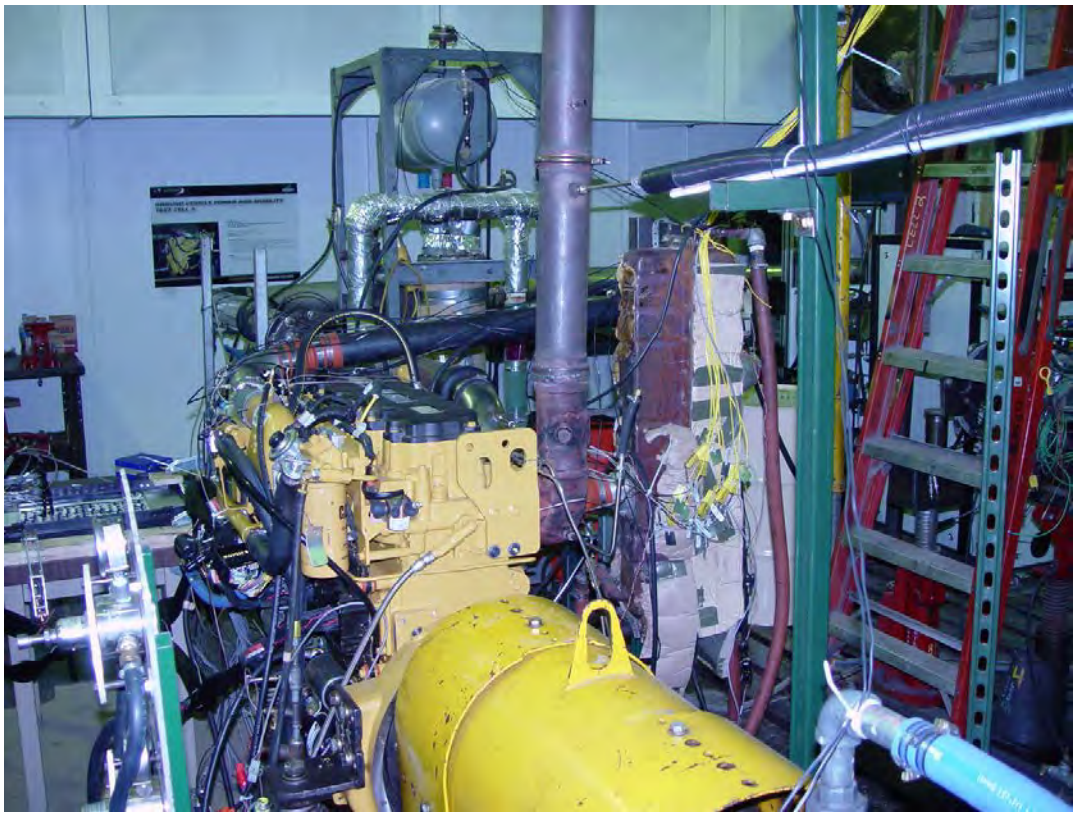

Figure 1. Instrumented CAT 7 engine in the TARDEC test cell.

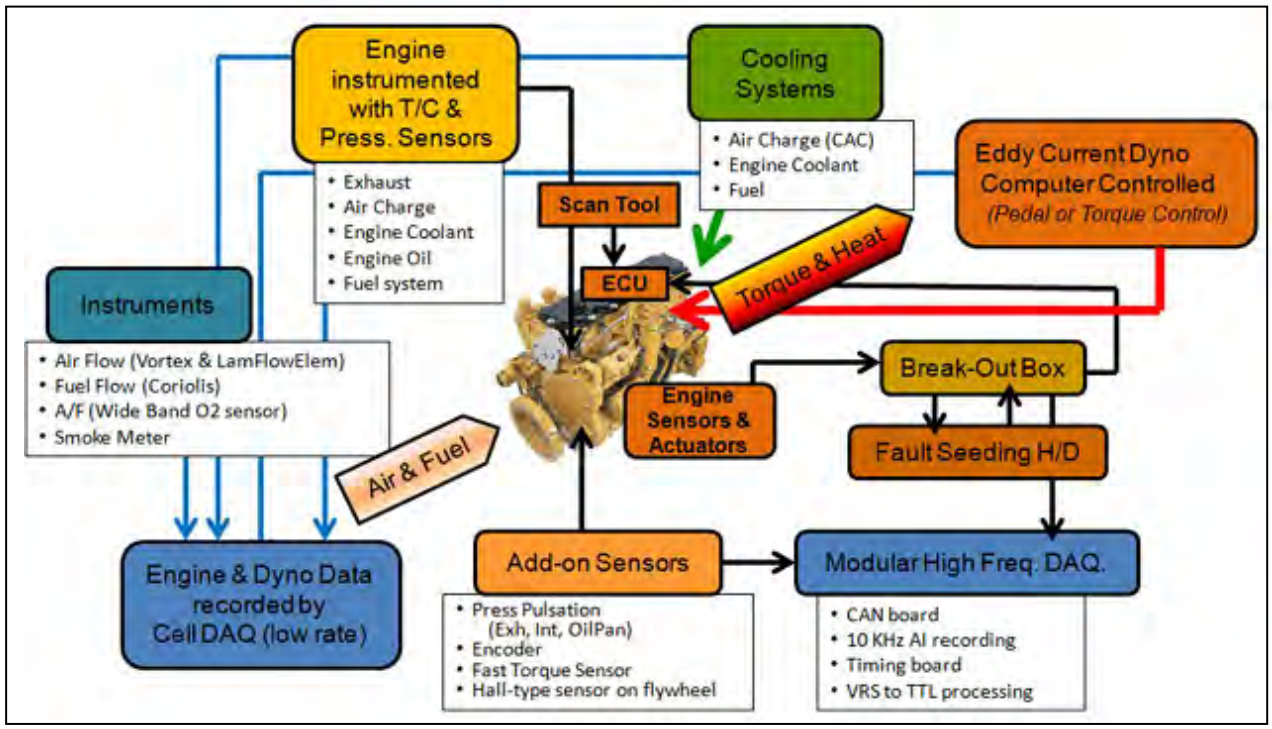

Figure 2. Engine control, instrumentation, and data flow.

Test runs were performed with various seeded faults and no fault cases. A test run consisted of running the engine through a stepwise sequence of designated speeds for a short time at each speed, as shown in figure 3; all with either no fault or a particular seeded fault. The engine speeds with associated duration were duplicated for all the tests. As can be seen, there are six speeds with duration of between 1 and 3 min each; the time duration at a given speed set point was not precisely controlled.

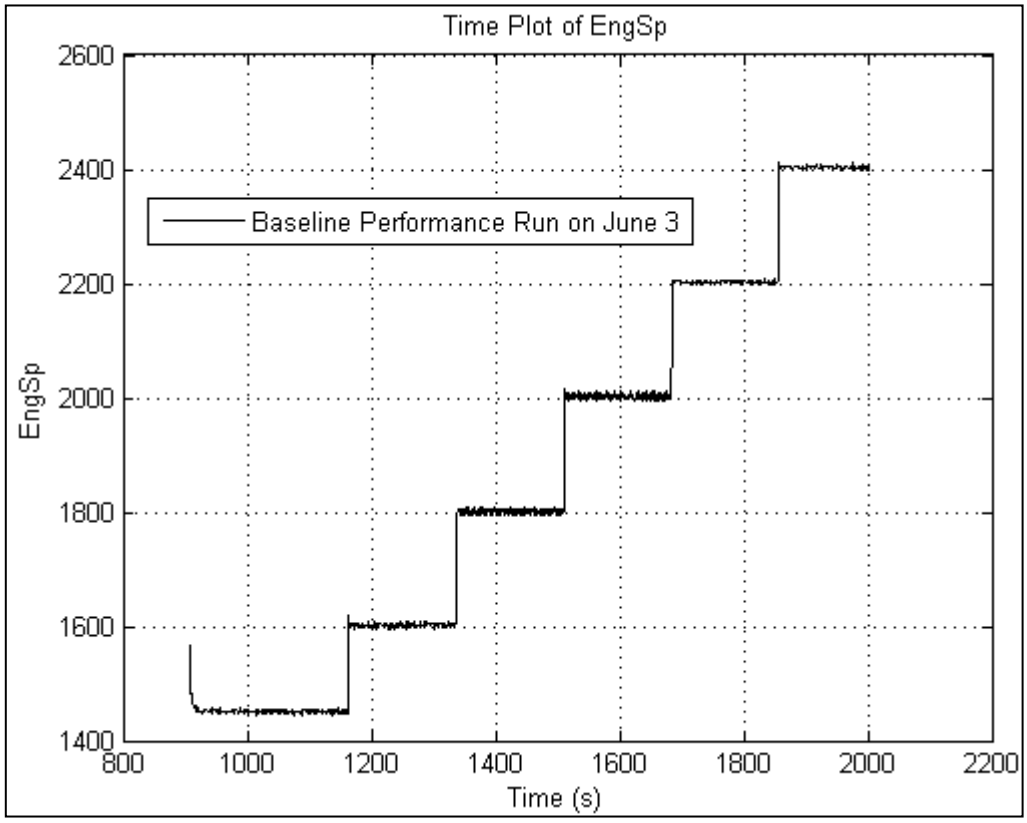

Figure 3. Typical stepped control of engine speed for a performance run.

3. Data for Analysis

The current focus is on the performance test data since these files have several baseline runs along with several seeded fault runs. Baseline runs are test sequences at the beginning of a test day in which there was no fault but the standard test sequence was followed, and as such are viewed as “healthy states.” Table 1 shows the 15 baseline runs that were identified. For principal component analysis (PCA) and autoassociative neural network based methods (AANN), training data is required; the first column of table 1 shows the runs that were selected for training (50% of the runs, using a random number generator). Table 2 shows the 33 seeded fault performance runs; however, three of these test runs are considered a baseline condition since their gain was

set to 1.0, which is the nominal value. As a note, several files contained more than one run, where the additional runs were various levels of the same fault type.

Table 1. Baseline performance runs.

Baseline Performance Test #	Date	MatLAB File Name	Run # in File	Train (0) or Test (1)
Training 1	May 27, 2011	PerfM3_JP8_May27_ext	1	0
Training 2	May 27, 2011	PerfM3_JP8_May27_ext	2	0
Test 1	June 1, 2011	Perf_Jun1_ext	1	1
Training 3	June 3, 2011	Perfor_Jun3_ext	1	0
Training 4	June 8, 2011	Perfor_Jun8_par	1	0
Test 2	June 10, 2011	Perfor_Jun10_ext	1	1
Training 5	June 15, 2011	Perfor_Jun15_ext	1	0
Test 3	June 16, 2011	Perfor_Jun16_ext	1	1
Test 4	June 22, 2011	Perfor_C_Jun22_ext	1	1
Test 5	June 29, 2011	Perfor_jun29_ext	1	1
Test 6	July 1, 2011	Perf_Jul1_ext	1	1
Training 6	July 6, 2011	Perfor_Jul6_ext	1	0
Training 7	July 8, 2011	Perfor_Jul8_ext	1	0
Test 7	July 27, 2011	Perfor_Jul27_ext	1	1
Test 8	August 3, 2011	Perfor_ext3_ext	1	1

Table 2. Seeded fault performance runs.

Test #	Date	MatLAB File Name	Fault Type	Run in File	Severity
9	May 27, 2011	PerfM3_IntRestr_May27_ext	IntakeAir Restric Test	1	Pos # 4
10	May 27, 2011	PerfM3_IntRestr_May27_ext	IntakeAir Restric Test	2	Pos # 6
11	June 8, 2011	PerfM3_OilP_Jun8_par	OilPress High Gain	1	Gain 1.0
12	June 8, 2011	PerfM3_OilP_Jun8_par	OilPress High Gain	2	Gain 0.7
13	June 8, 2011	PerfM3_OilP_Jun8_par	OilPress High Gain	3	Gain 1.3
14	June 10, 2011	PerfM3_AirChgT_Jun10_ext	Air Charge Temperature Increase	1	Increased by 20^{o}F
15	June 10, 2011	PerfM3_AirChgT_Jun10_ext	Air Charge Temperature Increase	2	Increased by 30^{o}F
16	June 10, 2011	PerfM3_AirChgT_Jun10_ext	Air Charge Temperature Increase	3	Increased by 50^{o}F
17	June 15, 2011	Perfor3_AirRestr_Jun15_ext	AirRestriction Low	1	Pos # 2
18	June 15, 2011	Perfor3_AirRestr_Jun15_ext	AirRestriction Low	2	Pos # 3
19	June 15, 2011	Perfor3_AirRestr_Jun15_ext	AirRestriction Low	3	Pos # 4
20	June 15, 2011	Perfor3_B_AirRestr_Jun15_ext	AirRestriction High	1	Pos #5
21	June 15, 2011	Perfor3_B_AirRestr_Jun15_ext	AirRestriction High	2	Pos #6
22	June 15, 2011	Perfor3_C_AirChgT_high_Jun15_ext	AirChgHigh	1	
23	June 15, 2011	Perfor3_C_AirChgT_high_Jun15_ext	AirChgHigh	2	
24	June 16, 2011	PerforM3_AirChg_low_Jun16_ext	AirCharge	1	
25	June 16, 2011	PerforM3_AirChg_low_Jun16_ext	AirCharge	2	
26	June 16, 2011	PerforM3_AirChg_low_Jun16_ext	AirCharge	3	
27	June 29, 2011	PerfM3_B_AirIntRes_Jun29_ext	IntRestriction	1	Pos #5
28	June 29, 2011	PerfM3_B_AirIntRes_Jun29_ext	IntRestriction	2	Pos #6
29	June 29, 2011	PerfM3_B_AirIntRes_Jun29_ext	IntRestriction	3	Pos #7
30	July 6, 2011	PerforM3_B_BoostG_Jul6_ext	Boost	1	Gain 0.85
31	July 6, 2011	PerforM3_B_BoostG_Jul6_ext	Boost	2	Gain 0.95
32	July 6, 2011	PerforM3_B_BoostG_Jul6_ext	Boost	3	Gain 1.00
33	July 13, 2011	PerforM3_ExhRestr_Jul13_ext	ExhRestr	1	60%
34	July 13, 2011	PerforM3_ExhRestr_Jul13_ext	ExhRestr	2	55%
35	July 13, 2011	PerforM3_ExhRestr_Jul13_ext	ExhRestr	3	50%
36	July 13, 2011	PerforM3_B_ExhRestr_Jul13_ext	ExhRestr	1	42%
37	July 13, 2011	PerforM3_B_ExhRestr_Jul13_ext	ExhRestr	2	46%
38	July 13, 2011	PerforM3_B_ExhRestr_Jul13_ext	ExhRestr	3	50%
39	August 3, 2011	PerforM3_InjPresG_ext3_ext	InjPress	1	Gain 1.0
40	August 4, 2011	PerforM3_InjPresG_ext3_ext	InjPress	2	Gain 0.9
41	August 5, 2011	PerforM3_InjPresG_ext3_ext	InjPress	3	Gain 1.1

From the data described, it was determined to initially work with the 45 signals from the CAN and dyno. Working with this set of low-cost sensor and CAN bus signals provides a path for a practical onboard implementation, and thus, is conducted first prior to considering vibration and other signals for developing health models. They also have been interpolated and aligned to the same 1 Sample/s acquisition rate, so were in a format that was ready to process. The CAN and dyno signals are identified in table 3. The 32 signals highlighted in orange were used in the analysis. The other 13 signals were not included because they are either operating conditions or have a low amount of variability.

Table 3. Signals recorded from CAN and dyno.

Signal #	Sensor Name
1	Time
2	EngSp
3	Load%
4	EngOilP
5	Boost
6	InjCtrlP
7	EngCoolT
8	IntManiAirT
9	Pedal%
10	ElPot
11	FuelRate
12	DesEngSp
13	NomFric%
14	Load@Sp

Signal #	Sensor Name
15	Fuel Flow
16	Speed
17	Torque
18	Throttle Pos
19	Lambda
20	AirFlow
21	BB-Torque-Sen
22	T-IntAirMani
23	T-aftCompr
24	CoolAftEngine
25	T-ExhB4Turbo1
26	T-ExhB4Turbo2
27	T-ExhStack
28	P-AirB4Mani
29	P-aftTurbo
30	P-ExhB4Turbo1
31	P-ExhB4Turbo2
32	'P-ExhStack'

Signal #	Sensor Name
33	T-OilGalley
34	P-OilGalley
35	ECM1-Boost
36	Sensor-Boost
37	ECM1-InjPres
38	Sensor-InjPres
39	ECM1-OilPres
40	Sensor-OilPres
41	ECM1-EngCoolT
42	Sensor-EngCoolT
43	ECM1-AirIntMani
44	Sensor-AirIntMani
45	Event

4. Modeling Approaches

Modeling approaches identified for the CAT 7 data, in order of complexity, included single parameter monitoring, correlation analysis, PCA monitoring methods, and AANN residual methods. Each of these methods has its advantages and disadvantages; table 4 lists each method along with important trade-offs. For the present study, only correlation analysis and PCA monitoring were undertaken. Single parameter modeling is very simple and can be done in real time; however, it requires extensive experience for setting thresholds for each variable, thus only an example of its application is presented at this time. The AANN method is well suited for this data and is being considered for future work.

Table 1. Advantages and disadvantages for each modeling approach identified for the CAT 7 data.

Method	Training Requirements	Threshold Setting	Positive/Negative
Single parameter monitoring	None	Experience or requires historical data	Simple but does not take into account relationship among variables
Correlation analysis	Multiple baseline data sets	Experience or requires historical data	Takes into account variable interaction but a less established method than PCA
PCA monitoring	Multiple baseline data sets	Established statistical limits	Well-established method but does not account for nonlinear interaction between variables
AANN method	Multiple baseline data sets and more computation	Statistical limits	Handles nonlinear variable interaction but requires more computation for training

4.1 Single Parameter Monitoring

Single parameter monitoring assumes that degradation in engine performance can be evaluated by one or more of the signals independently. It appears likely that this method can be applied to this data, but as mentioned, fault thresholds for a signal must be expertly set. Here we present only an example of how this method could be applied. First, it is noted that when the exhaust was restricted, the exhaust gas temperature is seen to be increased above baseline runs, as shown in figure 4. If it was known that an exhaust stack temperature above 1100 °F at an operating speed of 1450 RPM indicated that there was a blockage in the exhaust stack, then the faulted condition could be identified. Note that our example does not take into account what other variables might cause the exhaust stack temperature to increase.

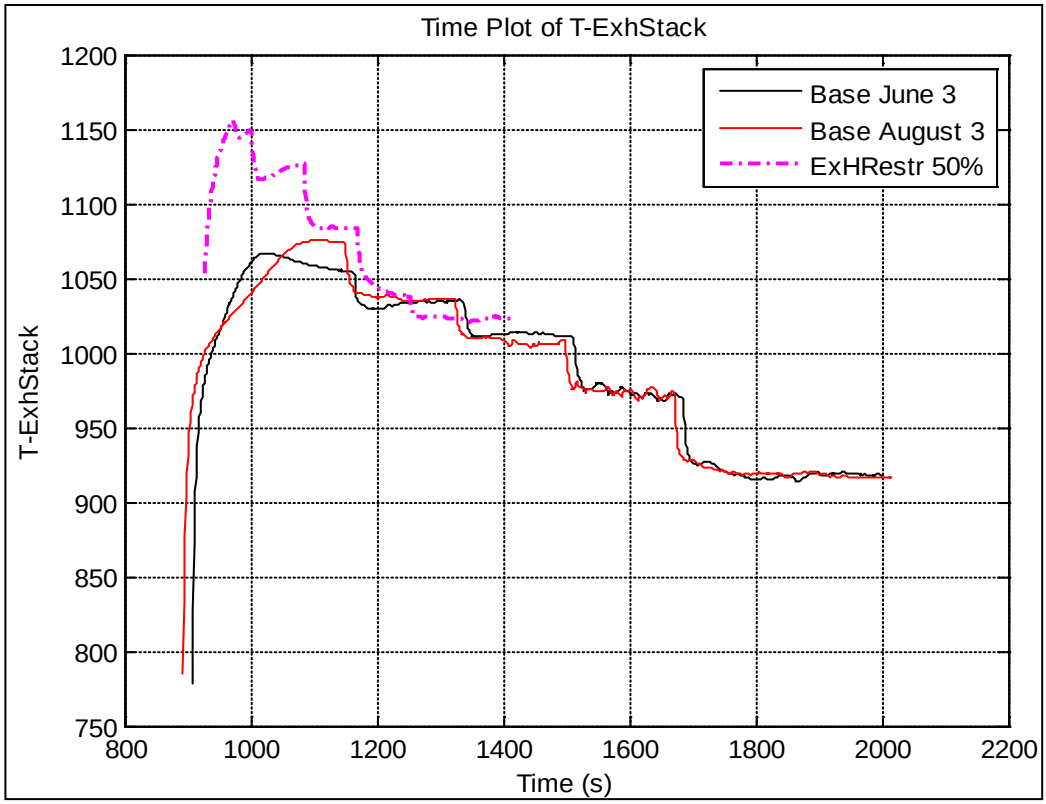

Figure 4. Exhaust stack temperature for an exhaust restriction of 50% compared with baseline data.

4.2 Correlation Analysis

The main concept of this approach is to look for correlation changes with respect to a template file. The seven baseline runs identified for training in table 1 were used as the template. The list of processing steps follows:

1. Select the regime and signal subset.
2. Perform the correlation matrix calculation for the baseline/template.
3. Perform the correlation matrix calculation for the test run.
4. Calculate the correlation difference matrix with respect to the template file.
5. Calculate a figure of merit (FOM) for the test run.
6. Health classification using a FOM threshold from the receiver operating characteristic curve (ROC curve)

The initial step includes the option of considering a particular operating regime or signal subset. A FOM value is calculated and provides a single indicator that can be used to assess the health condition of the engine. Additionally, searching the correlation difference matrix for maximum

changes provides a way of identifying which signals are contributing to the anomalous performance.

1. *Signal Subset and Operating Regime*

   The correlation analysis focused on the 32 signals highlighted in table 3. Only data that were in the operating regime with engine speed above 1500 RPM and engine load above 80% were considered; this included all of the standard run data, but cuts out deviations from the standard run that *are in* the actual data.

2. *Correlation and Correlation Difference Matrix*

   Correlation between two signals, s_i and s_j, is defined as the covariance between those two signals normalized by the variance of each signal $s_{si \text{ and } sj}$, as shown in equation 1.

$$r_{ij} = \frac{Cov(s_i, s_j)}{\sigma_{s_i} \sigma_{s_j}}, \tag{1}$$

   where covariance is defined as the expected value expression in the numerator (equation 2). Note that the correlation is calculated for each signal pair and provides a matrix that is N x N in size, where N is the number of signals (*2*).

$$r_{ij} = \frac{E\left[\left(s_i - \mu_{s_i}\right)\left(s_j - \mu_{s_j}\right)\right]}{\sigma_{s_i} \sigma_{s_j}} \tag{2}$$

   The correlation difference matrix is generated by subtracting the elements of the correlation matrix for the run data from the template and squaring it to produce a magnitude (equation 3).

$$d_{ij} = \left(r_{ij}\_Run - r_{ij}\_Template\right)^2 \tag{3}$$

   Correlation difference matrix plots are shown in figures 5 and 6. Figure 5 shows difference plots for two of the baseline runs (healthy) and indicates a low level of variation in healthy sets. Figure 6 is for an exhaust restriction run and shows distinct differences in correlation from the healthy template, particularly, in the T-ExhStack sensor compared with almost every other sensor.

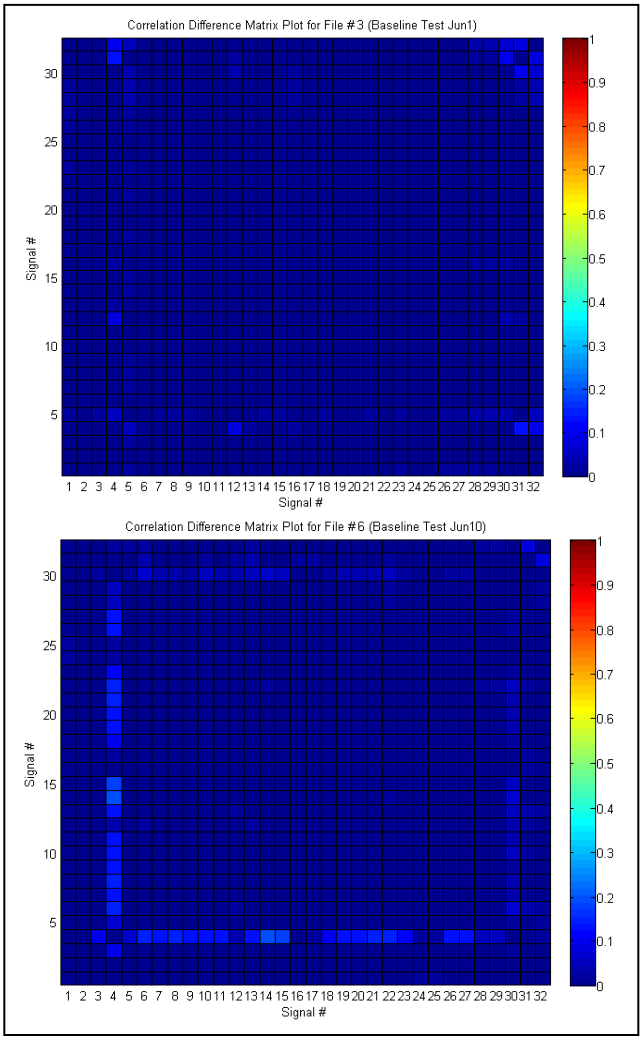

Figure 5. Correlation difference matrices for baseline runs (left) 3 and (right) 6.

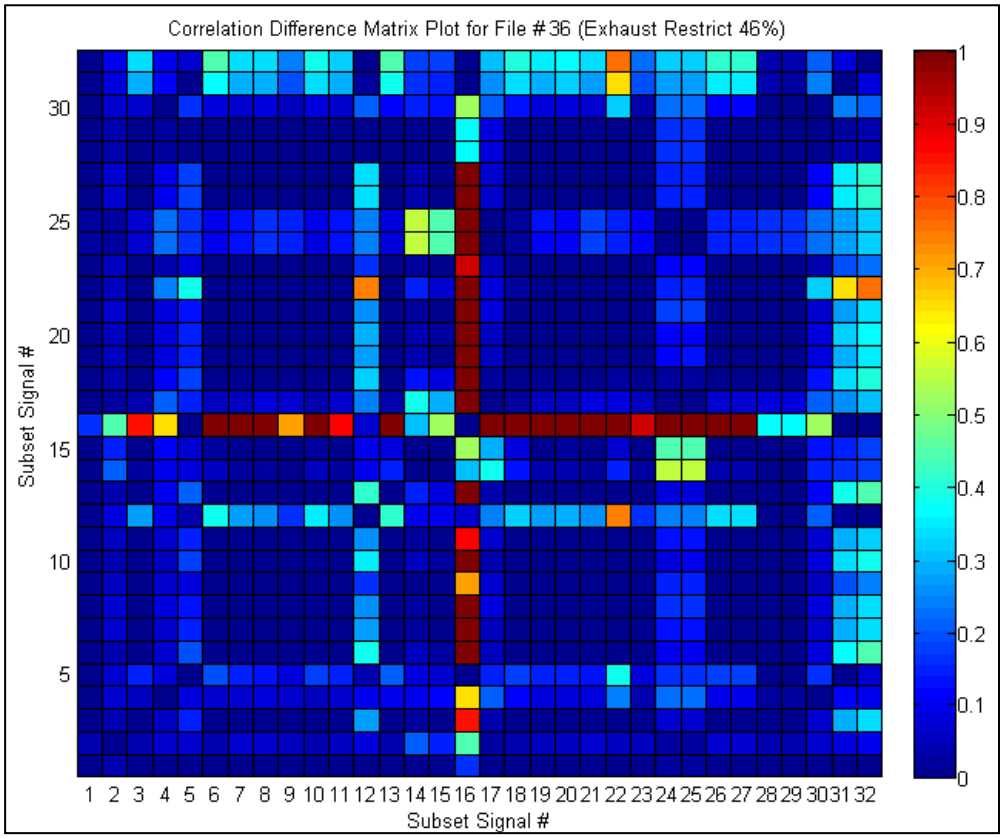

Figure 6. Correlation difference matrix for a 46% exhaust restriction (run 36).

FOM Calculation and Health Assessment

For each test run, a FOM value based on the correlation difference matrix was calculated. The FOM was defined as the summation of the values of the correlation difference matrix (equation 4):

$$FOM = \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} \tag{4}$$

To evaluate health of the system based on this FOM, there needs to be a threshold established above which the engine will be considered to be in a faulted state. The receiver operating characteristic curve (ROC curve) is a common way of showing classification/detection results as a function of false positives and false negatives as a threshold is varied (*3*). Figure 7 shows the ROC curve for the FOMs of this data. In this case, a threshold of approximately 44.6 for the FOM value offers the best trade-off. This provides a false alarm rate of 5.56% and a missed detection of 46.7%.

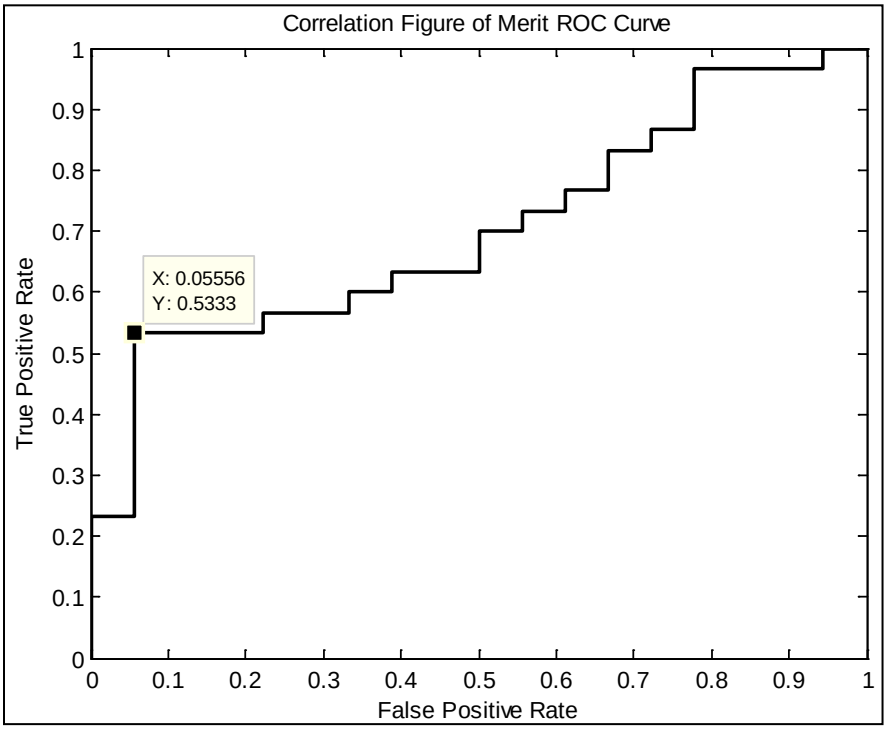

Figure 7. ROC curve for the correlation-based FOM.

The FOM health values are shown along with the threshold in figure 8. For the baseline runs, 10 of the 11 runs were classified correctly as healthy (9.1% false alarm rate). Note that three additional baseline runs were added and given a later test number since these are from the files in which the gain was varied, but for these runs the gain was 1.0. For the seeded fault runs, 14 of the 30 were misclassified as healthy. The missed detection rate is quite high and highlights that this method has difficulty in detecting lower levels of degradation (associated with the lower levels of particular faults). There is some implication, however, that some lower fault levels may not degrade engine performance, and it may not be correct to call them “unhealthy.” In general, this method is detecting the more severe induced faults but not the lower levels of the same fault.

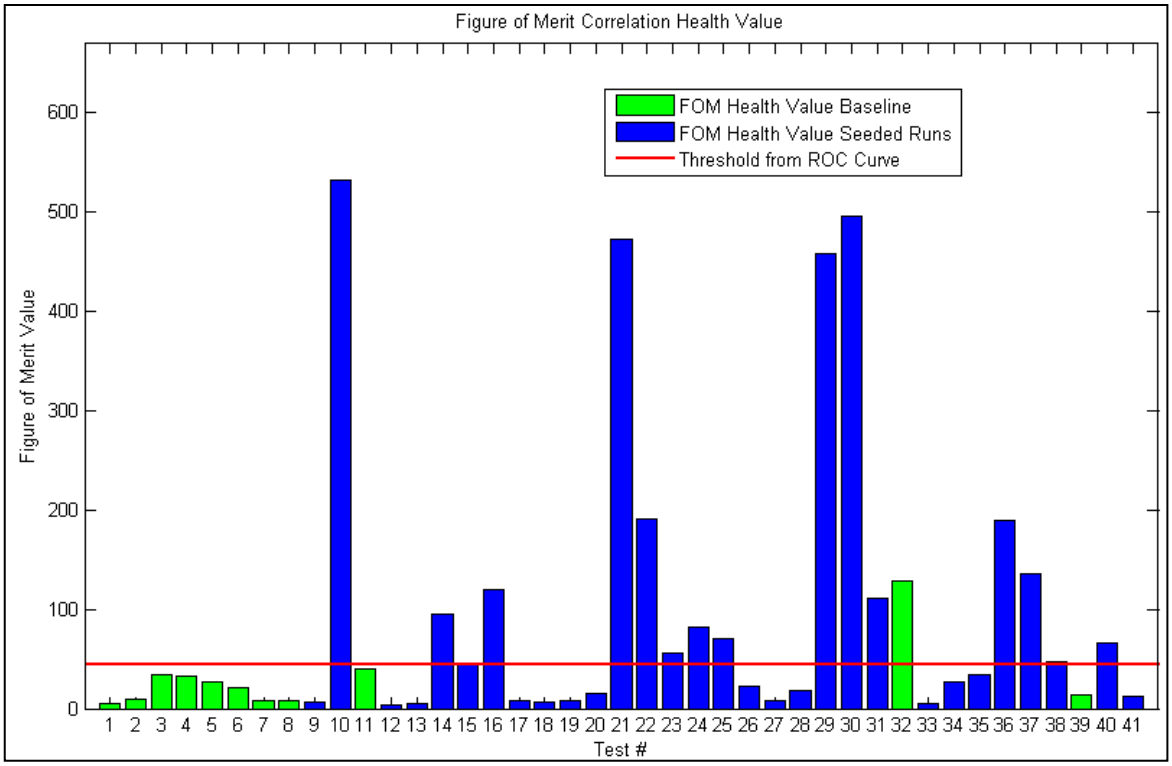

Figure 8. Correlation FOM for all runs The first eight are baseline runs and the remainder are by test number from table 2. The plot excludes training runs.

4.3 Principal Component Analysis

The primary concept is to extract useful information from the data set by projecting the data into a new set of orthogonal coordinates. PCA does this by performing an eigenvalue/eigenvector calculation on the covariance matrix (*4*). Its use for data analysis is diverse; for health monitoring, the application here, its use for dimension reduction (*5*) is applied and it is also used to calculate monitoring statistics (*6*). Specifically, the statistics T^2 and square prediction error (SPE) are calculated for the block of data in each operating regime. The mean of the health value in that block is used to decide on the health status based on thresholds derived from statistical theory. If any values are above those thresholds, contribution plots are used to further identify the source of the fault. Listed below are the steps that were followed. Details of the basic PCA calculations are omitted, and the reader is referred to references 4–8 for specifics on the use of PCA and formulae used:

1. Select the regime and signal subset
2. Normalize the data and calculate the covariance matrix for the training set.
3. Perform eigenvalue/eigenvector calculation of the covariance matrix.
4. Save PCA baseline models with normalization and eigenvalue/eigenvector information.
5. Normalize and project data from the monitored engine using baseline models.

f. Calculate the T^2 and SPE health statistics and calculate the mean of these for the block of data.

g. Calculate the top contributors for each fault.

The initial step is to select operating regimes and the signal list. The signals were the same as those used for the correlation analysis. Four regimes were selected, which represent steady state operating points in the performance test runs, as shown in table 5. To avoid transient effects, the first and last 20 s in a particular operating regime were not included in the calculations.

Table 2. Operating regimes for PCA analysis.

Regime No	Engine RPM	Engine Load	Pedal %
1	1620–1820	60–100	80–100
2	1820 – 2020	60–100	80–100
3	2020–2200	60–100	80–100
4	2220–2420	60–100	80–100

After calculating the principal components, its is seen that the first few principal components can explain most of the variation seen in the data. The typical approach for determining the number of principal components to retain is to look at the eigenvalues (ranked in decreasing order) and select the ones that explain a high percentage of the variability in the data. In this analysis, the percentage was set to 85%. As an example, in the case of Regime 4, the top principal component accounts for 37% of the variability in the data set and the first eight account for 85%. Figure 9 shows the decay in the variability for the first 10 signals; the variability continues to decay for the remaining 22.

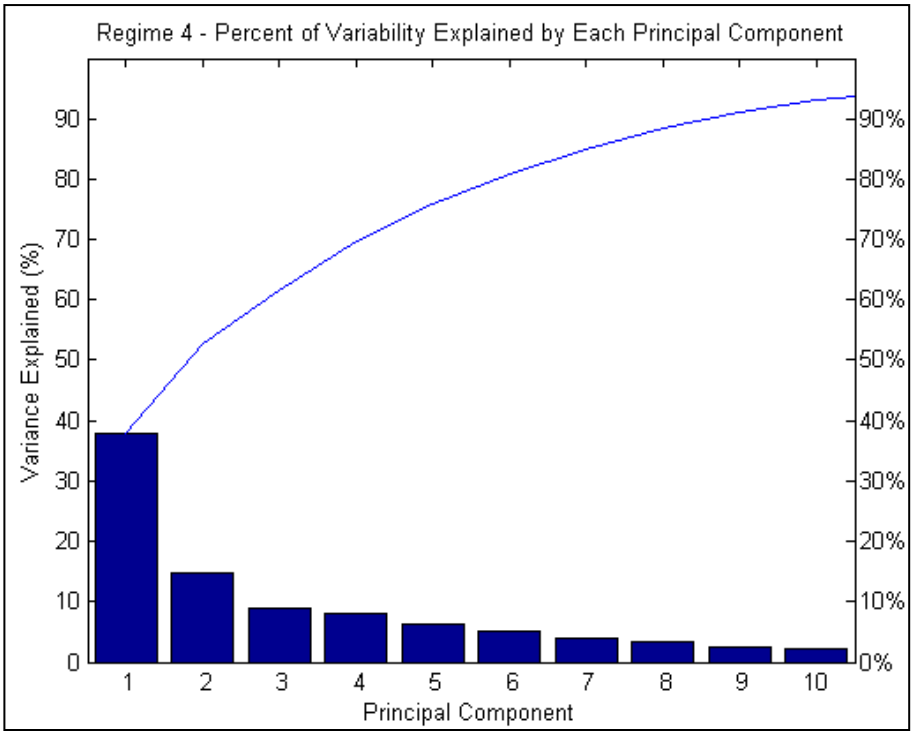

Figure 9. Percent of variability explained by each principal component (Regime 4).

Critical to the use of PCA for health monitoring are the calculation of the monitoring statistics T^2 and SPE, which are defined here with a full description available in reference 7. The T^2 calculation is similar to Mahalanobis distance, but is performed with the principal components, in our case, retained principal components, instead of the original data matrix (equation 5).

$$T^2 = \{u\}_{1xr}\left[\sum\right]_{rxr}^{-1}\{u\}_{1xr}^{T}, \tag{5}$$

where r is the number of retained principal components, $\{u\}_{1xr}$ are the retained principal components, and Σ_{rxr} are the retained eigenvalues

The residuals, *E*, are calculated, which are essentially the difference between the model and actual data values (equation 6):

$$\{E\}_{1xn} = \{x_n\}_{1xn} - \{u\}_{1xr}[P]_{nxr}^{T}, \tag{6}$$

where, *n* are the number signals, $\{x\}_{1xn}$ are the actual signal values, and $[P]_{nxr}^{T}$ are the retained eigenvectors.

SPE is the sum of the residuals (summed from residuals for each sensor) (equation 7):

$$SPE = \sum_{i=1}^{n} E_i \tag{7}$$

To make health assessments based on T^2 and SPE values, thresholds were calculated using commonly accepted techniques described below (*8*, *9*). The thresholds for T^2 is calculated using equation 8, where *r* is the number of principal components retained, *m* is the number of samples in the training data set, a is the confidence level, and *F* is the *F* value from the F-distribution table:

$$T_{\alpha}^{\ 2} = \frac{r(m-1)}{m-r} F_{r,m-r,\alpha}. \tag{8}$$

The SPE thresholds calculation is provided in equation 9. It is quite involved because the distribution of SPE is a summation of Chi-square distributions. See reference 9 for more details. C_a is the Z-value corresponding to a given confidence level (Normal Distribution Table) and l are the eigenvalues calculated from the training data set:

$$SPE_{\alpha} = \theta_1 \left[\frac{c_{\alpha}\sqrt{2\theta_2 h_0^{\ 2}}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right]^{\frac{1}{h_0}} \tag{9}$$

$$Where: \theta_i = \sum_{j=r+1}^{n} \lambda_j^i \ , \ \mathrm{i} = 1,2,3$$

$$and: h_0 = 1 - \frac{2\theta_1\theta_3}{3\theta_2^2}$$

PCA Results

Based on calculations for a 99% confidence level, the detection results are very good for most cases; the detection rates are shown in table 6. The results, in general, improve with engine speed. The false alarm rate is zero in all regimes except Regime 1. Even so, the two false alarms in Regime 1 had values that were only slightly above the detection threshold. Plots of SPE and T^2 for each regime along with the thresholds are presented in figures 10–17. Note that the *x*-axes are the original test numbers and that the *y*-axes are on a logarithmic scale (this is because of the large range in values). As a significant note, the majority of the missed detections were for air intake restrictions. For example, four of the five missed detections in Regime 4 occurred for the air intake seeded fault; this unfortunately significantly reduces the overall detection rate. Also of value, the monitoring statistics show a correlation with fault severity, in that faults of increasing severity had higher (more degraded) health values.

Table 3. Results of SPE and T^2.

Regime	Fault Detection Rate (SPE)	False Alarm Rate (SPE)	Fault Detection Rate (T^2)	False Alarm Rate (T^2)
1	80%	18.18%	73.33%	0.00%
2	80%	0.00%	70.00%	0.00%
3	83.33%	0.00%	73.33%	0.00%
4	83.33%	0.00%	83.33%	0.00%

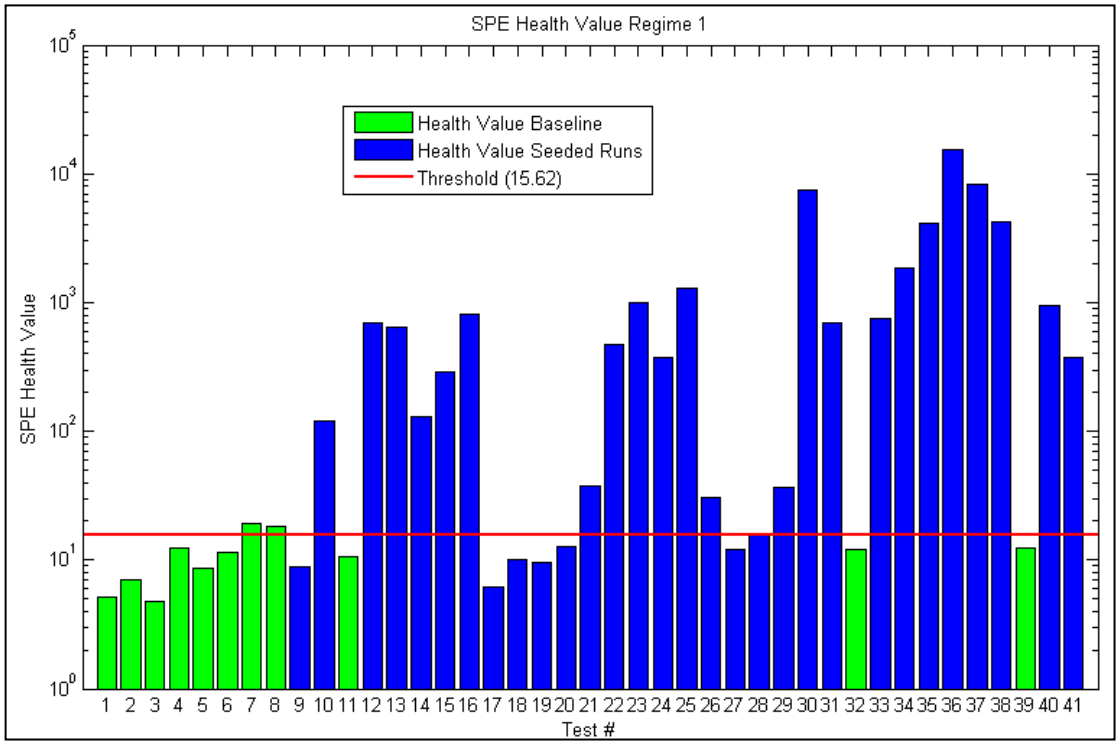

Figure 10. Regime 1 SPE health values for each run.

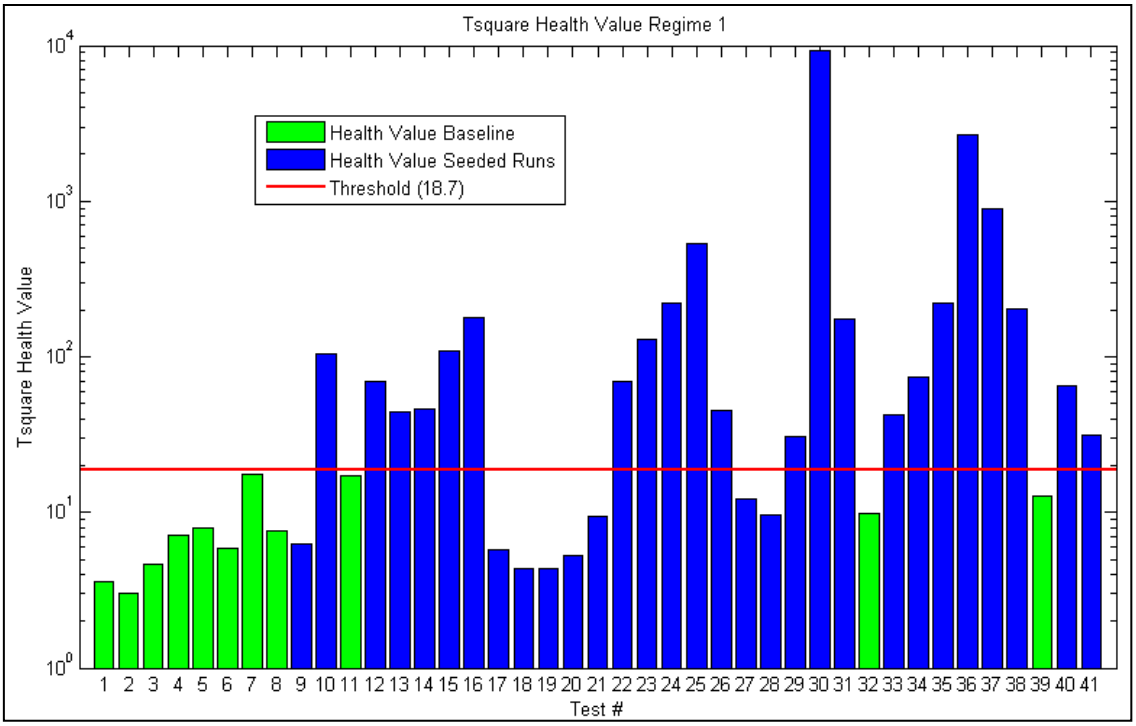

Figure 11. Regime 1 T^2 health values for each run.

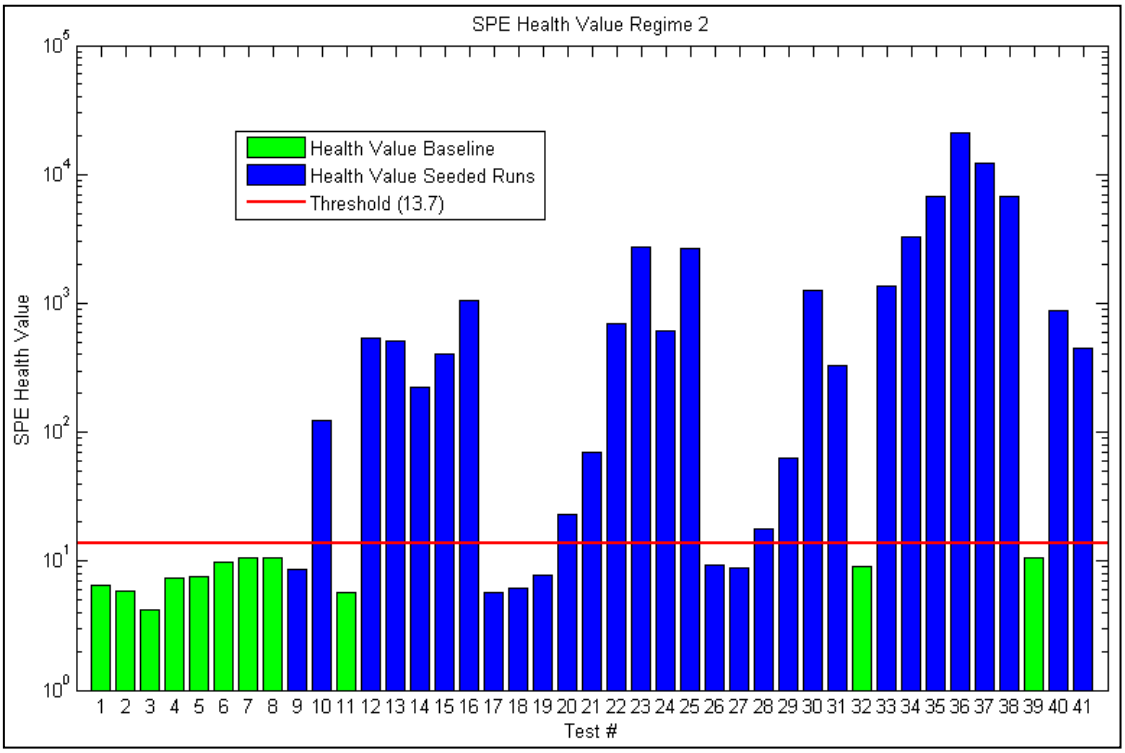

Figure 12. Regime 2 SPE health values for each run.

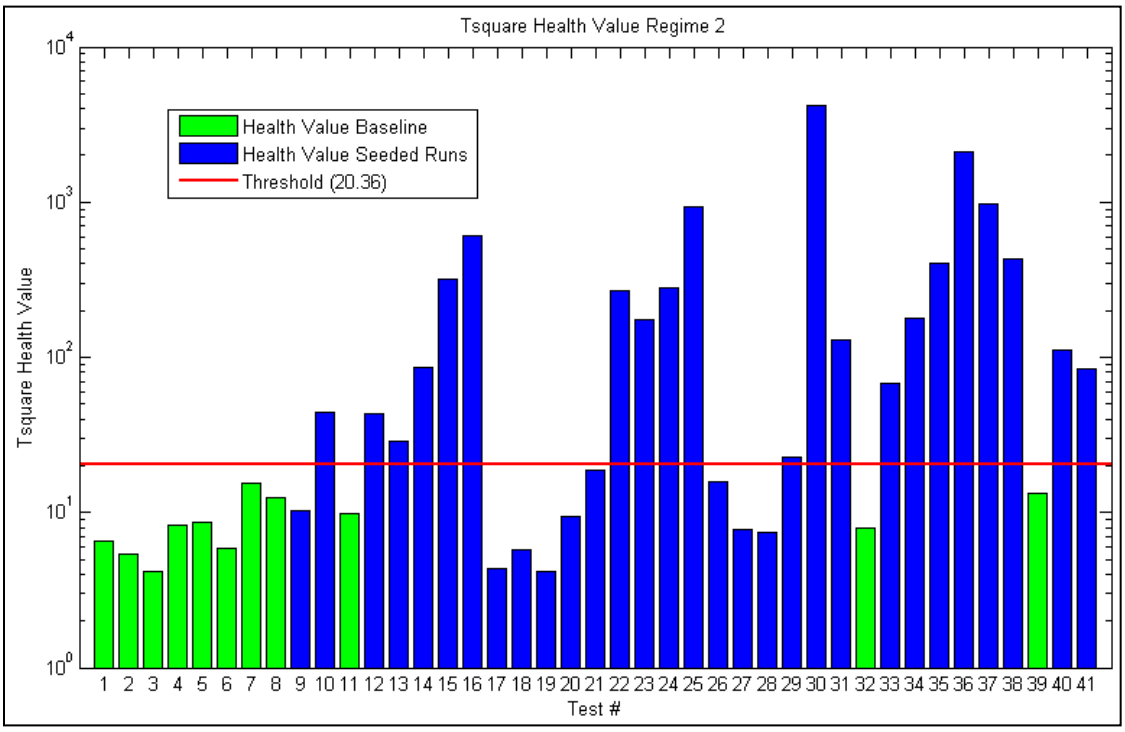

Figure 13. Regime 2 T^2 health values for each run

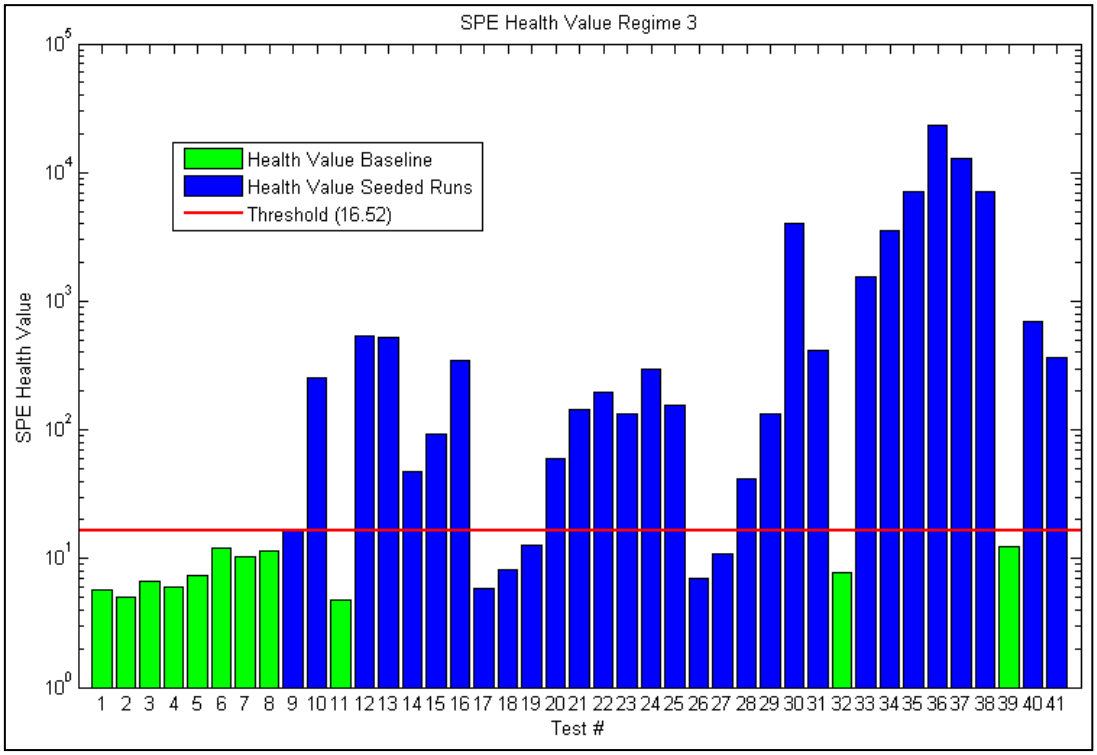

Figure 14. Regime 3 SPE health values for each run.

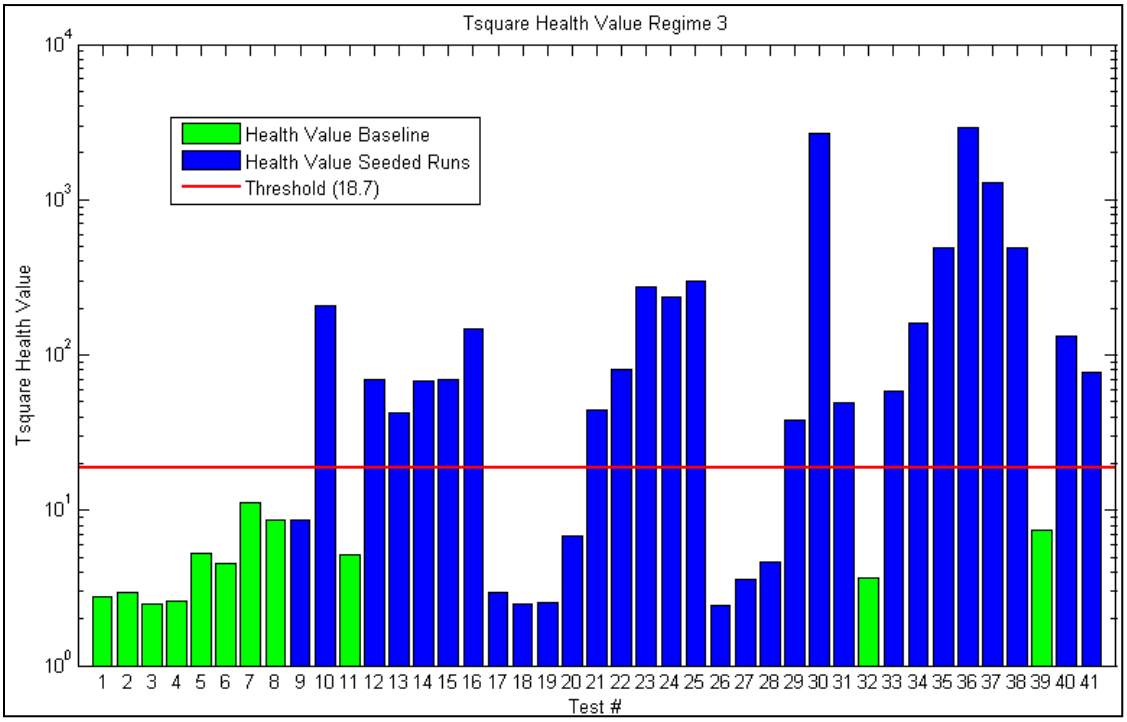

Figure 15. Regime 3 T^2 health values for each run.

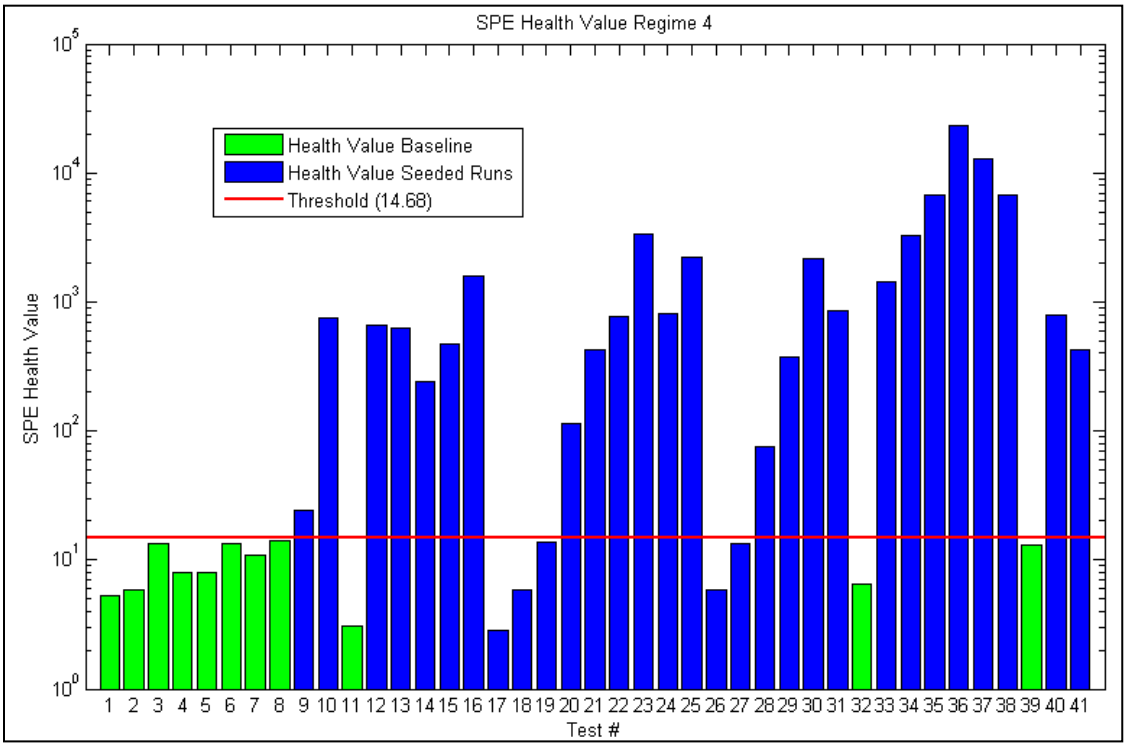

Figure 16. Regime 4 SPE health values for each run.

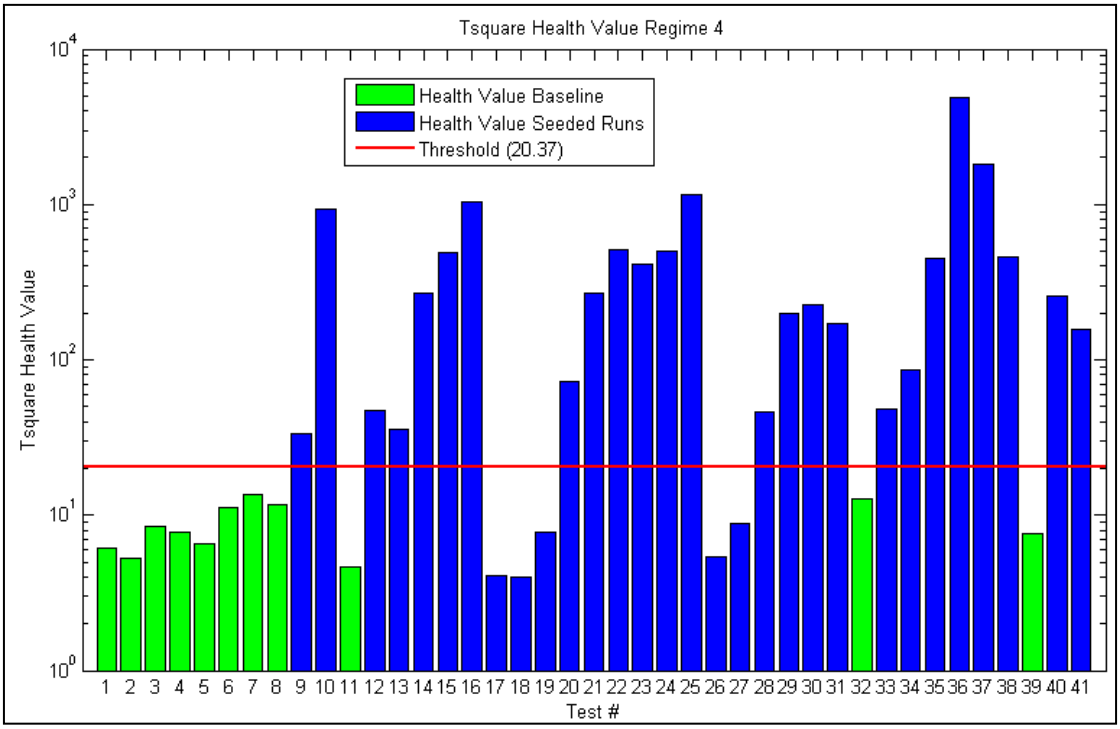

Figure 17. Regime 4 T^2 health values for each run.

It is of interest to know which signals had the greatest contribution for a particular fault. The T^2 and SPE contribution values for each of the signals are calculated using equations 7 and 8 (*9*). The results are presented in tables 7 and 8. An additional benefit from this study is that the contribution plot results could provide insight on what signals are important for single parameter monitoring.

$$\{T_{contribue}\}_k = \{u\}_{1xr}\left[\sum\right]_{rxr}^{-1}\left(\{x_{kcolumn}\}_{1x1}\left[P_{krow}\right]_{1xr}\right)^T \tag{7}$$

$$\{SPE_{contribute}\}_k = \frac{E_k^2}{\sum_{i=1}^{n} E_i^2} \tag{8}$$

where k varies from 1 to n for both equations 6 and 7.

In general, the top contributors for the SPE agree with a physical understanding of the engine and the faults that were seeded. For example, seeded faults with boost are showing that the boost sensor is the main contributor, and likewise seeded faults with exhaust restriction are showing that the exhaust pressure sensor is showing the most contribution. The contribution results from the T^2 are more difficult to interpret in some instances, for example, the boost sensor is the top contributor for the higher exhaust restriction faults. The faults that were induced by adjusting a sensor gain gave virtually the same contribution results for both T^2 and SPE; however, the mechanical faults based on restricting airflow resulted in different top contributors. Whether this

is always the case would require further experimentation and study. Figure 18 shows a particular run, 50% exhaust restriction, with the relative contributions of each sensor from SPE. In this case, the exhaust pressure sensor has a much higher contribution than the others, with the second largest contributor being the exhaust gas temperature.

Table 4. Signal contribution to fault detection from T^2 calculation.

Test #	MatLAB File Name	Fault Type	Severity	T Contribution 1	T Contribution 2
9	PerfM3_IntRestr_May27_ext	IntakeAir Restric Test	Pos # 4	'P-ExhB4Turbo2'	'P-ExhB4Turbo1'
10	PerfM3_IntRestr_May27_ext	IntakeAir Restric Test	Pos # 6	'ECM1-Boost'	'Sensor-Boost'
12	PerfM3_OilP_Jun8_par	OilPress High Gain	Gain 0.7	'ECM1-OilPres'	'EngOilP'
13	PerfM3_OilP_Jun8_par	OilPress High Gain	Gain 1.3	'ECM1-OilPres'	'EngOilP'
14	PerfM3_AirChgT_Jun10_ext	AirCharge Temp high Shift	Increased by 20^{0}F	'Sensor-AirIntMani'	'ECM1-AirIntMani'
15	PerfM3_AirChgT_Jun10_ext	AirCharge Temp high Shift	Increased by 30^{0}F	'IntManiAirT'	'Sensor-AirIntMani'
16	PerfM3_AirChgT_Jun10_ext	AirCharge Temp high Shift	Increased by 50^{0}F	'IntManiAirT'	'Sensor-AirIntMani'
17	Perfor3_AirRestr_Jun15_ext	AirRestriction Low	Pos # 2	'Sensor-InjPres'	'ECM1-InjPres'
18	Perfor3_AirRestr_Jun15_ext	AirRestriction Low	Pos # 3	'InjCtrlP'	'Sensor-InjPres'
19	Perfor3_AirRestr_Jun15_ext	AirRestriction Low	Pos # 4	'P-ExhB4Turbo2'	'P-ExhB4Turbo1'
20	Perfor3_B_AirRestr_Jun15_ext	AirRestriction High	Pos #5	'P-ExhB4Turbo2'	'P-ExhB4Turbo1'
21	Perfor3_B_AirRestr_Jun15_ext	AirRestriction High	Pos #6	'ECM1-Boost'	'Sensor-Boost'
22	Perfor3_C_AirChgT_high_Jun15_ext	AirChgHigh		'IntManiAirT'	'Sensor-AirIntMani'
23	Perfor3_C_AirChgT_high_Jun15_ext	AirChgHigh		'T-IntAirMani'	'Sensor-AirIntMani'
24	PerforM3_AirChg_low_Jun16_ext	AirCharge		'Sensor-AirIntMani'	'ECM1-AirIntMani'
25	PerforM3_AirChg_low_Jun16_ext	AirCharge		'ECM1-AirIntMani'	'IntManiAirT'
26	PerforM3_AirChg_low_Jun16_ext	AirCharge		'P-ExhB4Turbo2'	'InjCtrlP'
27	PerfM3_B_AirIntRes_Jun29_ext	IntRestriction	Pos #5	'P-ExhB4Turbo2'	'ECM1-EngCoolT'
28	PerfM3_B_AirIntRes_Jun29_ext	IntRestriction	Pos #6	'P-ExhB4Turbo2'	'T-ExhB4Turbo2'
29	PerfM3_B_AirIntRes_Jun29_ext	IntRestriction	Pos #7	'ECM1-Boost'	'Sensor-Boost'
30	PerforM3_B_BoostG_Jul6_ext	Boost	Gain 0.85	'ECM1-Boost'	'Boost'
31	PerforM3_B_BoostG_Jul6_ext	Boost	Gain 0.95	'Sensor-Boost'	'P-ExhB4Turbo2'
33	PerforM3_ExhRestr_Jul13_ext	ExhRestr	60%	'P-ExhB4Turbo2'	'P-ExhStack'
34	PerforM3_ExhRestr_Jul13_ext	ExhRestr	55%	'P-ExhB4Turbo2'	'P-ExhStack'
35	PerforM3_ExhRestr_Jul13_ext	ExhRestr	50%	'ECM1-Boost'	'Sensor-Boost'
36	PerforM3_B_ExhRestr_Jul13_ext	ExhRestr	42%	'Sensor-Boost'	'ECM1-Boost'
37	PerforM3_B_ExhRestr_Jul13_ext	ExhRestr	46%	'Sensor-Boost'	'ECM1-Boost'
38	PerforM3_B_ExhRestr_Jul13_ext	ExhRestr	50%	'ECM1-Boost'	'Sensor-Boost'
40	PerforM3_InjPresG_ext3_ext	InjPress	Gain 0.9	'Sensor-InjPres'	'T-ExhB4Turbo2'
41	PerforM3_InjPresG_ext3_ext	InjPress	Gain 1.1	'Sensor-InjPres'	'T-ExhB4Turbo2'

Table 5. Signal contribution to fault detection from SPE calculation.

Test #	MatLAB File Name	Fault Type	Severity	Q Contribution 1	Q Contribution 2
9	PerfM3_IntRestr_May27_ext	IntakeAir Restric Test	Pos # 4	'P-ExhB4Turbo2'	'P-AirB4Mani'
10	PerfM3_IntRestr_May27_ext	IntakeAir Restric Test	Pos # 6	'AirFlow'	'T-ExhB4Turbo2'
12	PerfM3_OilP_Jun8_par	OilPress High Gain	Gain 0.7	'ECM1-OilPres'	'EngOilP'
13	PerfM3_OilP_Jun8_par	OilPress High Gain	Gain 1.3	'ECM1-OilPres'	'EngOilP'
14	PerfM3_AirChgT_Jun10_ext	AirCharge Temp high Shift	Increased by 20^{o}F	'T-IntAirMani'	'IntManiAirT'
15	PerfM3_AirChgT_Jun10_ext	AirCharge Temp high Shift	Increased by 30^{o}F	'T-IntAirMani'	'IntManiAirT'
16	PerfM3_AirChgT_Jun10_ext	AirCharge Temp high Shift	Increased by 50^{o}F	'T-IntAirMani'	'IntManiAirT'
17	Perfor3_AirRestr_Jun15_ext	AirRestriction Low	Pos # 2	'T-ExhB4Turbo2'	'P-ExhStack'
18	Perfor3_AirRestr_Jun15_ext	AirRestriction Low	Pos # 3	'Torque'	'P-ExhB4Turbo1'
19	Perfor3_AirRestr_Jun15_ext	AirRestriction Low	Pos # 4	'P-ExhB4Turbo2'	'AirFlow'
20	Perfor3_B_AirRestr_Jun15_ext	AirRestriction High	Pos #5	'P-ExhB4Turbo2'	'AirFlow'
21	Perfor3_B_AirRestr_Jun15_ext	AirRestriction High	Pos #6	'AirFlow'	'P-ExhB4Turbo2'
22	Perfor3_C_AirChgT_high_Jun15_ext	AirChgHigh		'T-IntAirMani'	'T-ExhB4Turbo2'
23	Perfor3_C_AirChgT_high_Jun15_ext	AirChgHigh		'Sensor-AirIntMani'	'ECM1-AirIntMani'
24	PerforM3_AirChg_low_Jun16_ext	AirCharge		'T-IntAirMani'	'ECM1-EngCoolT'
25	PerforM3_AirChg_low_Jun16_ext	AirCharge		'Sensor-AirIntMani'	'ECM1-EngCoolT'
26	PerforM3_AirChg_low_Jun16_ext	AirCharge		'P-ExhB4Turbo2'	'P-aftTurbo'
27	PerfM3_B_AirIntRes_Jun29_ext	IntRestriction	Pos #5	'P-ExhB4Turbo2'	'AirFlow'
28	PerfM3_B_AirIntRes_Jun29_ext	IntRestriction	Pos #6	'P-ExhB4Turbo2'	'AirFlow'
29	PerfM3_B_AirIntRes_Jun29_ext	IntRestriction	Pos #7	'AirFlow'	'P-ExhB4Turbo2'
30	PerforM3_B_BoostG_Jul6_ext	Boost	Gain 0.85	'ECM1-Boost'	'Sensor-Boost'
31	PerforM3_B_BoostG_Jul6_ext	Boost	Gain 0.95	'ECM1-Boost'	'Sensor-Boost'
33	PerforM3_ExhRestr_Jul13_ext	ExhRestr	60%	'P-ExhStack'	'P-ExhB4Turbo2'
34	PerforM3_ExhRestr_Jul13_ext	ExhRestr	55%	'P-ExhStack'	'P-ExhB4Turbo2'
35	PerforM3_ExhRestr_Jul13_ext	ExhRestr	50%	'P-ExhStack'	'T-ExhStack'
36	PerforM3_B_ExhRestr_Jul13_ext	ExhRestr	42%	'P-ExhStack'	'T-ExhStack'
37	PerforM3_B_ExhRestr_Jul13_ext	ExhRestr	46%	'P-ExhStack'	'T-ExhStack'
38	PerforM3_B_ExhRestr_Jul13_ext	ExhRestr	50%	'P-ExhStack'	'T-ExhStack'
40	PerforM3_InjPresG_ext3_ext	InjPress	Gain 0.9	'Sensor-InjPres'	'ECM1-InjPres'
41	PerforM3_InjPresG_ext3_ext	InjPress	Gain 1.1	'Sensor-InjPres'	'ECM1-InjPres'

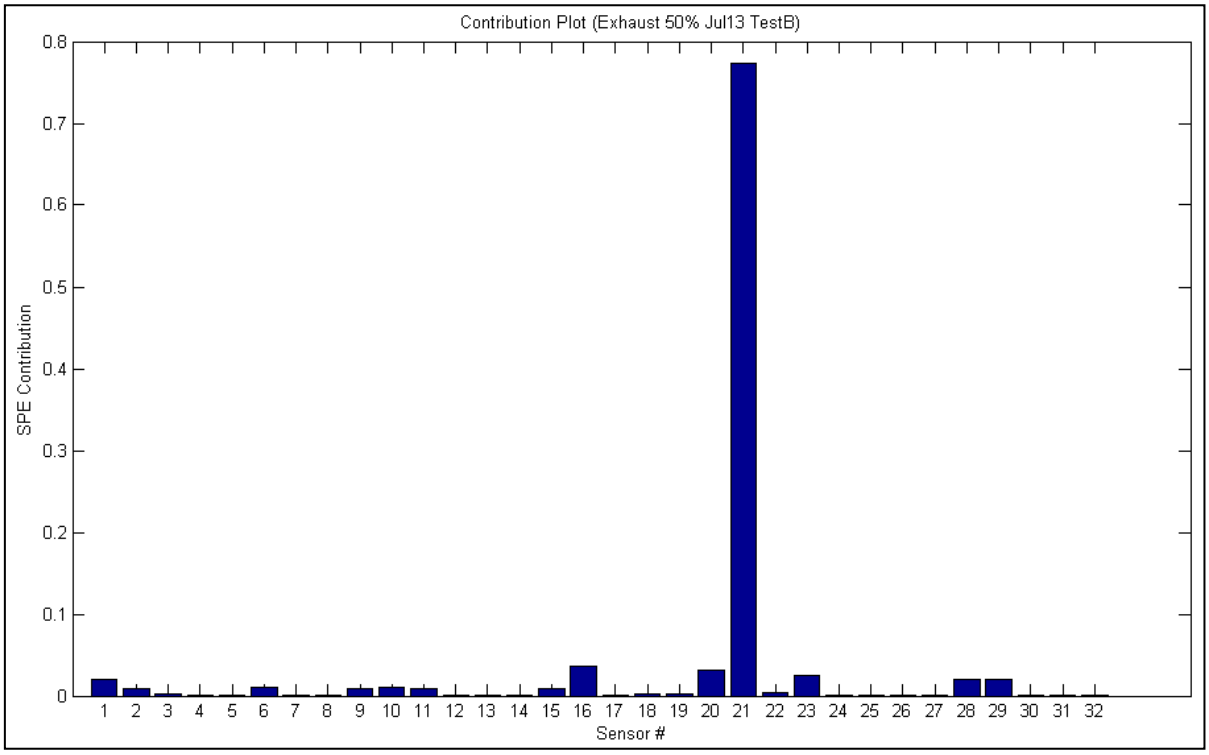

Figure 18. SPE contribution plot, showing the relative contribution of each signal (50% exhaust restriction).

5. Discussion

Several items of interest were discovered in this preliminary study including relative performance of the methods evaluated as well as salient characteristics of the results. First, when faults were applied, differences in the sensor outputs were detected. Therefore, it is reasonable to assume that single parameter modeling can be applied to health assessment of the engine. Again, the caveat of requiring expert knowledge to set signal thresholds inhibits our use of this method at this time. Although such information is difficult to come by, if it were to become available, then this method would be simple to implement. Second, it is seen that PCA is better suited to detect faults in this data than correlation analysis. The primary drawback of correlation analysis here is its inability to detect lower-level faults. There are several items of interest with PCA on this data set. It is a curiosity that the results improve with engine speed and we speculate that the effects of the fault are exacerbated as the speed increases. It is a matter for further study why the faults that were induced by adjusting a sensor gain gave the same contribution results for both T^2 and SPE, while the mechanical induced faults gave different top contributors. Finally, there is the matter that both methods could not detect the faults in all but the highest states of intake air restriction. At this time, we can only suggest that the lower states do not appear to have a significant effect on engine performance. This emphasizes a point regarding the nature of this testing; although named seeded faults, the runs may be more accurately described as perturbations in operating variables and are not faults in the traditional sense. These perturbations may or may not adversely affect engine performance. With this in mind, our work is on the detectability of the perturbations; and whether or not they are critical to the actual "health" of the engine is uncertain.

6. Conclusion and Recommendations

Single parameter monitoring, correlation analysis, and PCA with two independent statistics—T^2 and SPE—all show applicability to this problem. As discussed, single parameter monitoring can be pursued further if thresholds in signals become available. Encouragingly, both PCA statistics and correlation analysis detect the majority of the faults. PCA by far outperformed correlation analysis, and between the two any further work should focus on PCA. For the PCA method, various model refinements can be done, such as adjusting how many principal components to retain, using a smaller sensor subset, or incorporating the analog and Penn State data in the analysis. Finally, it is recommended to evaluate the data using nonlinear PCA; AANN is a proven way of implementing this approach (*11*). The motivation for using the AANN approach is based on the belief that some of the sensors in the engine have a nonlinear relationship/correlation.

7. References

1. MIS 2000, Final Report, "A Dynamic Neural Network Approach to CBM", Cooperative Agreement **W911-09-2-0036.**

2. Montgomery, D. C.; Runger, G. C.; Hubele, N. F. *Engineering Statistics*, Second edition, New York, Wiley, 2001.

3. Zweig, M.; Campbell, G. Receiver-Operating Characteristics (ROC) Plots: A Fundamental Evaluation Tool in Clinical Medicine. *Clinical Chemistry 39* (4), pp. 561–577

4. Abdi, H.; Williams, L. J. Principal component analysis, Wiley Interdisciplinary Reviews. *Computational Statistics* **2010**, *2*, 433–450.

5. Worden, K.; Staszewski, W. J.; Hensman, J. J. Natural computing for mechanical system research: A tutorial overview. *Mechanical Systems and Signal Processing* **2011**, *25*, 4–111.

6. Kano, M.; Nagao, K.; Hasebe, S.; Hashimoto, I.; Ohno, H.; Strauss, R.; Bakshi, B. Comparison of statistical process monitoring methods: application to the Eastman challenge problem. *Computers and Chemical Engineering* **2000**, *24*, 175–181.

7. Gallagher, V. B.; Wise, R. M.; Butler, S. W.; White, D. D.; Barna, G. G. Development and benchmarking of multivariate statistical process control tools for semiconductor etch process; improving robustness through model updating. In *proceedings of ADCHEM 97*, pp. 78–83, 1997.

8. Antory, D. Application of a data-driven monitoring technique to diagnose air leaks in an automotive diesel engine: A case study. *Mechanical Systems and Signal Processing* **2007**, *21* (2), 795–808.

9. Jackson, J. E.; Mudholkar, G. S. Control procedures for residuals associated with principal component analysis. *Technometrics* **2979**, 331–349.

10. Kano, M.; Hasebe, S.; Hashimoto, I.; Strauss, R.; Bakshi, B. R.; Ohno, H. Contribution plots for fault identification based on the dissimilarity of process data. In *AIChe Annual Meeting*, Technical Presentation, 2000.

11. Kramer, M. A. Nonlinear principal component analysis using autoassociative neural networks. *AIChe Journal 37* (2), 233–243.

List of Symbols, Abbreviations, and Acronyms

AANN	auto-associative neural network based methods
ARL	U.S. Army Research Laboratory
CAN	controller-area network
DAQ	data acquisition system
dyno	dynamometer
FOM	figure of merit
MIS	Millennium Integrated Services
P&D	prognostics and diagnostics
PCA	principal component analysis
Penn State	Pennsylvania State University
ROC	receiver operating characteristic curve
SPE	square prediction error
TARDEC	U.S. Army Tank and Automotive Research, Development and Engineering Center

NO. OF COPIES	ORGANIZATION
1 (PDF ONLY)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA 8725 JOHN J KINGMAN RD STE 0944 FORT BELVOIR VA 22060-6218
1	DIRECTOR US ARMY RESEARCH LAB IMNE ALC HRR 2800 POWDER MILL RD ADELPHI MD 20783-1197
1	DIRECTOR US ARMY RESEARCH LAB RDRL CIO LL 2800 POWDER MILL RD ADELPHI MD 20783-1197
1	DIRECTOR US ARMY RESEARCH LAB RDRL CIO MT 2800 POWDER MILL RD ADELPHI MD 20783-1197
4	DIRECTOR US ARMY RESEARCH LAB ATTN: RDRL SER E ANDREW J BAYBA DAVID N SIEGEL KWOK TOM CANH LY 2800 POWDER MILL RD ADELPHI MD 20783-1197
3 (PDF ONLY)	DIRECTOR US ARMY RESEARCH LAB ATTN: RDRL VTM M HAILE ATTN: RDRL VTM A GHOSHAL ATTN: RDRL VTM M MURUGAN BLDG 4603 APG MD 21005

NO. OF COPIES	ORGANIZATION
1 (PDF ONLY)	NASA GLENN US ARMY RESEARCH LAB ATTN RDRL VTP B DYKAS BLDG 23 RM W121 CLEVELAND OH 44135-3191
1 (PDF ONLY)	NASA GLENN US ARMY RESEARCH LAB ATTN RDRL VTP H DECKER BLDG 23 RM W121 CLEVELAND OH 44135-3191
3 (PDF ONLY)	US ARMY TARDEC ATTN RDTA RS C BECK ATTN RDTA RS K FISHER ATTN RDTA RS S HUSSAIN MS# 204 6501 E 11 MILE RD WARREN MI 48397-5000
1 (PDF ONLY)	USAMSAA ATTN T. S. KILBY 392 HOPKINS RD APG MD 21005
TOTAL: 17 (10 ELEC, 7 HCS)	